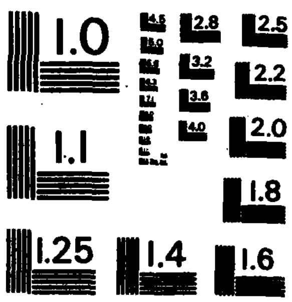
AD-A134 491 A LINGUISTIC OVERVIEW OF THE NIGEL TEXT GENERATION GRAMMAR(U) UNIVERSITY OF SOUTHERN CALIFORNIA MARINA DEL REY INFORMATION SCIENCES INST W C MANN OCT 83 1/1

UNCLASSIFIED ISI/RS-83-9 F49620-79-C-0181 F/G 5/7 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

ISI Reprint Series
ISI/RS-83-9
October 1983

12

University of Southern California

William C. Mann

A Linguistic Overview of the Nigel Text Generation Grammar

INFORMATION SCIENCES INSTITUTE
ISI
213/822-1511
4676 Admiralty Way / Marina del Rey / California 90292

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER ISI/RS-83-9	2. GOVT ACCESSION NO. AD-A134491	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A Linguistic Overview of the Nigel Text Generation Grammar		5. TYPE OF REPORT & PERIOD COVERED Research Report
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) William C. Mann		8. CONTRACT OR GRANT NUMBER(s) F49620-79-C-0181
9. PERFORMING ORGANIZATION NAME AND ADDRESS USC/Information Sciences Institute 4676 Admiralty Way Marina del Rey, CA 90292		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Air Force Office of Scientific Research Building 410, Bolling Air Force Base Washington, DC 20332		12. REPORT DATE October 1983
		13. NUMBER OF PAGES 16
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) ----------		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

This document is approved for public release; distribution is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

----

18. SUPPLEMENTARY NOTES

This report is a reprint of a paper that appears in the proceedings of *The Tenth LACUS Forum*, held in Quebec in August, 1983. *The Tenth LACUS Forum* is published by Hornbeam Press.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

artificial intelligence, choice experts, computer generation of text, inquiry semantics, knowledge representation methods, natural language, Nigel, Penman, semantics of English, systemic grammar, text generation

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

(OVER)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. ABSTRACT (continued)

Recent text generation research resembles recent research in synthesis of vaccines. The research is designed to construct entities which previously arose naturally. This constructive approach creates practical and theoretical benefits.

Our text generation research has produced a large systemic English grammar, which is embedded in a computer program. This grammar, which is called Nigel, generates sentences. It is controlled by a semantic stratum which has been added to the basic systemic framework.

This paper describes the program, which also is called Nigel. It identifies augmentations of various precedents in the systemic framework, and it indicates the current status of the program. The paper has a dual focus. First, on Nigel's processes, it describes the methods Nigel uses to control text to fulfill a purpose by using its new semantic stratum. Second, concerning Nigel's interactions with its environment, it shows reasons why Nigel is easily embedded in a larger experimental program.

Although the paper does not focus on Nigel's syntactic scope, that its scope is non-trivial is indicated by the fact that all of the sentence and clause structures of this abstract are within that syntactic scope.

ISI Reprint Series
ISI/RS-83-9
October 1983

University of Southern California

William C. Mann

A Linguistic Overview of the Nigel Text Generation Grammar

Availability Codes
Avail and/or
Dist Special
A1

INFORMATION SCIENCES INSTITUTE ISI
213/822-1511
4676 Admiralty Way / Marina del Rey / California 90292

This research was supported by the Air Force Office of Scientific Research contract No. F49620-79-C-0181. The views and conclusions contained in this document are those of the author and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Office of Scientific Research of the U.S. Government.

ISI Reprint Series

This report is one in a series of reprints of articles written by ISI research staff and published in professional journals and conference proceedings. For a complete list of ISI reports, write to

Document Distribution
USC/Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90291
USA

Contents

1. Progress in Immunology: Synthetic Vaccines

In February of this year, *Scientific American* described a breakthrough in medical science, the laboratory synthesis of vaccines against flu and other virus diseases [Lerner 83]. Ever since vaccines were conceived, research on vaccines and immunology has focused on natural substances and their effects. Now it has become possible to synthesize vaccines to fight many common diseases.

In synthesizing vaccines, scientists had to supplement the established methodologies. In particular, they had to develop methods of vaccine ***construction*** to supplement existing methods of vaccine ***identification*** and ***description***.

When the work on synthesis began, there was a substantial amount of evidence to suggest that the task would be overwhelmingly complex, and, in a practical sense, impossible. The *Scientific American* article describes how they did it. They worked with models at the molecular level, a finer level of detail than that of most of the preceding work. One of the key stages in the process came when they discovered how to cause synthesized vaccine elements to express the many abstract attributes by which vaccines had previously been described (without reference to the molecular level.)

> "[We found that] ...synthetic peptides can mimic the distinctions revealed by serologic studies; in designing synthetic vaccines, one will be able to take advantage of serologic evidence..."

The key evidence of success, of course, is that the vaccines work. Out of voluminous and detailed reasoning about molecules, their shapes, and their interactions, come chemicals that *actually prevent the infections that they are supposed to prevent*. The success of the vaccines validates the work at the molecular level, and also validates the particular theories at higher levels on which it was based. When a synthesized vaccine is ineffective, it points to inadequacies in the supporting theories.

So, work aimed at creating synthetic vaccines has also created a powerful new tool for scientific inquiry. Its worth can be fully justified on its scientific benefits alone, or on its practical benefits alone.

2. Penman: Constructive Research in Linguistics

The work described in this paper is another variety of scientific work in a constructive methodology. In this case, the items being constructed are texts rather than vaccines, and the test of effectiveness involves reading the texts rather than resisting a disease. Despite the differences, the work on synthetic vaccines helps us to understand current work on ***text generation***.

In text generation, the focal task is to create a text, in fluent natural language (often English), in response to some particular need. The trick is to do so using only the explicit knowledge of language specified by one's linguistic theories, rather than using the skills of a person. Research of this sort has been carried out in a scattered fashion for over a decade, generally using a computer as the site of synthesis and as the repository of the most detailed level of theory [Davey 79, Mann

[1] This report is a reprint of a paper that appears in the proceedings of *The Tenth LACUS Forum*, held in Quebec in August, 1983. *The Tenth LACUS Forum* is published by Hornbeam Press.

81, Mann & Moore 80, McKeown 82, Mann & Matthiessen 83a]. Computers here fill the role of molecular genetics for the vaccine work: they are a relatively new technology, enabling the work in a practical sense but not in any way fundamental to it.

Text generation research uses a constructive methodology, but it is vitally dependent on prior descriptive work. Because it is a constructive approach, voluminous detail is required; descriptive research, with its heavy reliance on processes of abstraction, eliminates details. Descriptive research can proceed in several independent directions to unreconciled conclusions; constructive research requires extensive reconciliation of the contributing theories. Just as for the vaccines, the best validation of the constructive theory is the observable effectiveness of the synthesized results.

Several years ago, we began work on a new text generation system, named Penman, designed to write texts a few paragraphs long. A previous round of research had led to a computer program which was able to write a limited range of two-paragraph texts, but which had several serious shortcomings, especially in the narrow rigidity of its grammar.

We therefore wanted to include in Penman a significant, linguistically justified grammar. Now, several years later, we have such a grammar, named Nigel after Halliday's learner [Halliday 75]. This paper passes over the parts of Penman devoted to invention, to text planning and to retrospective improvement of the text, and concentrates entirely on Nigel, the grammar.

3. Nigel: Penman's Grammar

The grammatical framework of Penman is the Systemic framework, begun by Michael Halliday in the late 1950's.[1] It draws on a wide range of prior work, including [Halliday 76, Hudson 76, Halliday & Martin 81, Fawcett 80, Berry 75, Berry 77, Halliday & Hasan 76] and others. In order to reconcile the various fragments of grammar and to augment them, all of the prior work has been altered in some way, sometimes by a simple notational shift, and sometimes by thorough re-representation.

I will describe Nigel in a series of stages, in effect working backward through the generation process from the level of lexical items to the level of conditions in Nigel's environment which affect the particular text produced. The whole discussion will be about Nigel's role in generating single sentences, because Penman is designed to plan text down to the sentence level, including the relations that each sentence will express, and then to have Nigel execute each sentence plan independently.

3.1 Lexicon

Nigel's lexicon is deliberately oversimplified, because we felt that the limiting technical problems were elsewhere. It is a lexicon of independent lexical items, without morphology. The lexicon is well elaborated for lexical features: there are over 100 distinct lexical features, and the lexicon has items representing over 500 distinct combinations of lexical features. However, these figures are not particularly significant, since the lexicon has not been extensively developed or tested.

[1] The work on Nigel would not have been possible without the active participation of Christian Matthiessen, and the participation and past contributions of Michael Halliday and other systemicists.

3.2 Realization

Nigel builds syntactic structures by a set of activities usually called "realization" in the systemic framework. They are distinct from activities which specify the characteristics of a syntactic unit, formally termed ***grammatical features***. Each syntactic unit is first developed as a set of grammatical features, which realization converts to a *syntactic structure*. All of the control over what is built is exercised during the creation of the feature set; there is no optionality or syntactic variability in realization.

In Nigel, <u>all realization is realization of single features.</u> Each grammatical feature may have one or more ***realization statements*** associated with it, each consisting of a ***realization operator*** and a number of operands. Each realization statement makes some change or introduces some restriction on the structure being produced.

There are three groups of realization operators: those that build structure (in terms of grammatical functions), those that constrain order, and those that associate features with grammatical functions.

1. The realization operators which build structure are ***Insert***, ***Conflate***, and ***Expand***. By repeated use of the structure-building functions, the grammar is able to construct sets of ***function bundles***, also called ***fundles***. None of these operators are new to the systemic framework.
2. Realization operators which constrain order are ***Partition***, ***Order***, ***OrderAtFront***, and ***OrderAtEnd***. Partition constrains one function (hence one fundle) to be realized to the left of another, but does not constrain them to be *adjacent*. Order constrains just as Partition does, and in addition constrains the two to be realized adjacently. OrderAtFront constrains a function to be realized as the leftmost among the daughters of its mother, and OrderAtEnd symmetrically as the rightmost. Of these, only Partition is new to the systemic framework.
3. Realization operators that associate features with functions are ***Preselect***, which associates a grammatical feature with a function (and hence with its fundle); ***Classify***, which associates a ***lexical feature*** with a function; ***OutClassify***, which associates a lexical feature with a function in a preventive way; and ***Lexify***, which forces a particular lexical item to be used to realize a function. Of these, OutClassify and Lexify are new, taking up roles previously filled by Classify. OutClassify restricts the realization of a function (and hence fundle) to be a lexical item which <u>does</u> <u>not</u> bear the named feature. This is useful for controlling items in exception categories (e.g., reflexives) in a localized, manageable way. Lexify allows the grammar to force selection of a particular item without having a special lexical feature for that purpose. It is Preselect which makes the grammar recursive, since Preselect requires choosing a particular grammatical feature in a lower-ranked pass through the grammar.

In addition to these realization operators, there is a set of ***Default Function Order Lists***. These are lists of functions which will be ordered in particular ways by Nigel, provided that the functions on the lists occur in the structure and that the realization operators have not already ordered those functions. A large proportion of the constraint of order is performed through the use of these lists.

Published descriptions of the use of ordering in the systemic framework leave substantial room for interpretation. Programming the ordering operations of Nigel has convinced us that

systemic ordering is in fact a fairly complex matter. The (as yet unpublished) ordering algorithms of Nigel constitute a definite and testable proposal for the meanings of the realization operators for ordering.

3.3 Choice: Systems and Gates

Nigel has systems of alternatives, called ***systems*** as in the systemic tradition (to the confusion of the computational tradition.) The alternatives are grammatical features. Each system also has an ***entry condition***, a logical expression of grammatical features. The entry condition must be satisfied in order to enter the system, i.e., to have the set of alternatives available for choice. When a system has been entered, one of the alternative features must be chosen.

In addition to the systems there are ***Gates.*** A gate can be thought of as an entry condition which activates a particular grammatical feature, without choice. These grammatical features are used just as those chosen in systems. Gates are most often used to provide a feature to be realized, in response to a collection of features.[2]

3.4 Choosing

The systemic literature has many discussions of the oppositions of language, the direct alternations represented in systems. There is much less discussion of which alternative is most suitable in particular cases. Of course for text generation, making good individual choices is an essential activity, and so there must be some representation of how choices in the grammar are to be made.[3]

In order to specify explicitly how choices are made, a new definitional stratum has been added to systemic notation. For each system, a formally defined process called a ***chooser*** or ***choice expert*** is created. Each such process consists of steps, potentially of several kinds. The principal kinds of steps are information gathering, discrimination between kinds of conditions, and choice. When a system is entered, the corresponding chooser process is executed, yeilding a choice among the system's alternatives.

By defining choosers in this way, we can make explicit what particular choices depend upon, and we can examine whether particular natural examples conform to the conditions of choice which have been defined.

The activity of defining choosers often reveals regularities (or irregularities) which the grammar does not represent. Several choosers may depend in the same way on the same determinative condition. Or a notion such as markedness may turn out to represent very different

---

[2] There are no realization operations which depend on more than one feature, and no rules corresponding to Hudson's function realization rules. The gates facilitate eliminating this category of rules, with a net effect that the notation is more homogeneous.

[3] We find it useful to equivocate on the term "grammar," using it sometimes to represent the usual varieties of entities represented by systemic notation, and sometimes to include as well the mechanisms described below. Context will always disambiguate.

conditions in its various grammatical systems. Defining choosers typically leads to local refinement of the grammar, along with strengthened justification for the particular form used.

3.5 Inquiry

It would be possible to allow choosers to have some sort of unrestricted access to the knowledge which surrounds them, but this would be unsatisfactory as theory and unmanageable as a practical text generation resource. Instead, Nigel has a very simple, highly restricted method for choosers to gain access to the information they need. Choosers gain information to guide their work only by issuing ***inquiries*** stated in a simple ***inquiry language.***

The boundary of the grammar separates two nearly independent symbol systems. Outside of the boundary, in the ***environment***, there is knowledge of what needs to be said, including both general knowledge and the text plan. Inside the boundary are grammatical features, grammatical function symbols, chooser definitions, system and gate definitions, and realization statements. All of these are beyond the reach of the environment and cannot be designated for manipulation by it. Conversely, the symbol system outside of the environment is not directly available to the grammar. When the grammar needs a symbol to use in some later inquiry, such as a designation of the agent of a process so that it can inquire whether the agent is multiple, it asks for a temporary symbol for this purpose. These symbols are discarded once the unit has been built, and the separation of symbol systems is thereby maintained.

(Lexical items are exceptions to these remarks about symbol system separation. The choosers assume that associations are maintained between the relevant knowledge and lexical items, so that, for example, the grammar can elicit a set of denotationally appropriate terms as candidates to serve as the head of a nominal group.)

This way of defining and using an inquiry language has important practical and theoretical benefits. It permits development of the grammar and its semantics while avoiding two traps:

1. defining the grammar's semantics in terms of particular conventions of knowledge representation;
2. defining the grammar's semantics relative to particular syntactic structures, rather than to their functional import.

Avoiding the first of these traps is particularly important if Nigel is to be used in other artificial intelligence research projects. Since Nigel can be independent of particular knowledge representation formalisms, the rapid progress being made in knowledge representation will not render Nigel's definitions obsolete.

3.6 Environment

The environment is defined as the collection of symbol systems beyond the grammar's boundary. It consists of two rather different collections of information. Informally, they are

1. the ***Knowledge Base***: information which existed prior to the demand for which text is being generated; and

2. the *Text Plan*: information which is created in response to that demand.

Nigel leans heavily on both. It presumes that the text plan contains definite intentions about the ideational, interpersonal, and textual functions of the unit being generated; much of the ideational information comes from the knowledge base.

4. The Inquiry Stratum as a Semantics

Although definitions differ widely, the term "semantics" is usually used to represent some sort of specification of correspondence between elements of a linguistic system and elements of another system distinct from it. Taken this way, there are two senses in which the inquiries of Nigel constitute a semantics.

First, given a grammar of a language, including choosers, the collection of inquiry operators used in the choosers constitutes a specification of what can be expressed in syntactic structure. For example, multiplicity, intention to emphasize, and time precedence are identifiably expressed in Nigel's grammar of English. And we can also say, on the basis of the collection of inquiry operators, that English tense is indifferent to the contrast between moments and intervals. In this sense, the collection of inquiry operators provides a semantics of the collection of syntactic structures.

In the second sense, given the particular choosers, systems, and realization statements of a grammar, we can construct the mapping from particular conditions, i.e., particular collections of environmental responses to inquiries, to strings of symbols which they yield. This is a semantics of the grammar of particular utterances.

Note that, in both cases, a semantics of the grammar is specified, rather than a semantics of the language as a whole. Lexical aspects are specified in only a very rudimentary way, and the semantics above the level of the largest grammatical unit is likewise only slightly constrained. These limitations can be regarded as advantages, because they provide a principled factoring of a very complex field of inquiry.

5. State of Development

The generation mechanisms of Nigel have been programmed and tested. Choosers for about two thirds of its 200-odd systems have been defined. Whenever a new cluster of choosers is defined, there is an inevitable reexamination of the systems of that region, and of their justification. As a result, Nigel as a whole is evolving toward an increasingly homogeneous grammar in a fairly consistent definitional style.

When there are choosers for all of Nigel's systems, many new tests will become possible. They will involve generating units on demand, attempting to imitate natural examples, and characterizing syntactic units by the demands for which they were produced. Such tests, while vital and informative, are local to the grammar. They cannot show how adequate the grammar is as an element of a text generator.

We look forward to eventually mating Nigel with a programmed text planner, and later with programmed search processes (for invention) and text improvement processes as well. Only then can Penman be tested as a synthetic vaccine is tested--by *judging its operational results*. Seeing the products of other text generators, we expect that Penman will eventually generate very high quality text, and that the process of defining the generator will be filled with exciting and informative research.[4]

[4]Additional information on particular aspects of the work can be found in related reports and publications: Penman's design: [Mann 83a], Chooser definition: [Mann 82], Nigel's processes: [Mann & Matthiessen 83b], Inquiry semantics: [Mann 83b], Extended examples of Nigel's operation: [Mann & Matthiessen 83a].

References

[Berry 75] Berry, M., *Introduction to Systemic Linguistics: Structures and Systems*, B. T. Batsford, Ltd., London, 1975.

[Berry 77] Berry, M., *Introduction to Systemic Linguistics: Levels and Links*, B. T. Batsford, Ltd., London, 1977.

[Davey 79] Davey, A., *Discourse Production*, Edinburgh University Press, Edinburgh, 1979.

[Fawcett 80] Fawcett, R. P., *Exeter Linguistic Studies*. Volume 3: *Cognitive Linguistics and Social Interaction*, Julius Groos Verlag Heidelberg and Exeter University, 1980.

[Halliday 75] Halliday, M. A. K., *Learning How to Mean*, Edward Arnold, 1975.

[Halliday 76] Halliday, M. A. K., *System and Function in Language*, Oxford University Press, London, 1976.

[Halliday & Hasan 76] Halliday, M. A. K., and R. Hasan, *Cohesion in English*, Longman, London, 1976. English Language Series, Title No. 9.

[Halliday & Martin 81] Halliday, M.A.K., and J. R. Martin (eds.), *Readings in Systemic Linguistics*, Batsford, London, 1981.

[Hudson 76] Hudson, R. A., *Arguments for a Non-Transformational Grammar*, University of Chicago Press, Chicago, 1976.

[Lerner 83] Lerner, Richard A., "Synthetic Vaccines," *Scientific American* 248, (2), February 1983.

[Mann 81] Mann, W. C., "Two discourse generators," in *The Nineteenth Annual Meeting of the Association for Computational Linguistics*, Sperry Univac, 1981.

[Mann 82] Mann, W. C., *The Anatomy of a Systemic Choice*, USC/Information Sciences Institute, Marina del Rey, CA, Technical Report RR-82-104, October 1982. To appear in Discourse Processes

[Mann 83a] Mann, William C., *An Overview of the Penman Text Generation System*, USC Information Sciences Institute, Marina del Rey, CA 90291., Technical Report RR-83-114, 1983. To appear in the 1983 AAAI Proceedings.

[Mann 83b] Mann, William C., "Inquiry Semantics: A Functional Semantics of Natural Language Grammar," in *Proceedings of the First Annual Conference*, Association for Computational Linguistics, European Chapter, September 1983.

[Mann & Matthiessen 83a] Mann, W. C., and C. M. I. M. Matthiessen, *Nigel: A Systemic Grammar for Text Generation*, USC/Information Sciences Institute, RR-83-105, February 1983. The papers in this report will also appear in a forthcoming volume of the *Advances in Discourse Processes Series*, R. Freedle (ed.): *Systemic Perspectives on Discourse: Selected Theoretical Papers from the 9th International Systemic Workshop* to be published by Ablex.

[Mann & Matthiessen 83b] Mann, William C. and Christian M. I. M. Matthiessen, *An Overview of the Nigel Text Generation Grammar*, USC Information Sciences Institute, Marina del Rey, CA 90291., Technical Report RR-83-113, 1983.

[Mann & Moore 80] Mann, W. C., and J. A. Moore, *Computer as Author--Results and Prospects*, USC/Information Sciences Institute, RR-79-82, 1980.

[McKeown 82] McKeown, K.R., *Generating Natural Language Text in Response to Questions about Database Structure*, Ph.D. thesis, University of Pennsylvania, 1982.